

1        **In the Claims**

2        Claims 1, 13, 14, 21, 37, 69, 71, 81 are amended.

3        Claims 62-63 are cancelled.

4        Claims 1-61 and 64-96 remain in the application and are listed as follows:

5  
6        1.        (Currently Amended) A method for delivering software via a  
7 network comprising:

8                describing one or more software extensions using a hierarchical language,  
9 the extensions being configured for incorporation on a client, said describing  
10 defining one or more manifests containing at least one list of files comprising an  
11 extension; and

12                delivering the one or more manifests to the client via the network, the one  
13 or more manifests being configured for use in downloading the software  
14 extensions via the network, at least some of the extensions being downloadable by  
15 streaming extension files to the client in a manner that enables a user to begin to  
16 interact with the extension sooner than if the user had to wait for the entire  
17 extension to load.

18  
19        2.        (Original) The method of claim 1, wherein the one or more  
20 manifests are configured to assist in organizing delivery of individual files listed in  
21 the one or more manifests.

22  
23        3.        (Original) The method of claim 1, wherein the one or more  
24 manifests are configured to assist in validating individual files listed in the one or  
25 more manifests.

1  
2 4. (Original) The method of claim 1, wherein the one or more  
3 manifests are configured to assist in updating individual files listed in the one or  
4 more manifests.

5  
6 5. (Original) The method of claim 1, wherein the one or more  
7 manifests describe individual file locations.

8  
9 6. (Original) The method of claim 1, wherein the one or more  
10 manifests contain individual hashes for one or more of the listed files.

11  
12 7. (Original) The method of claim 1, wherein the one or more  
13 manifests contain download directives for downloading the listed files.

14  
15 8. (Original) The method of claim 1, wherein the one or more  
16 manifests are defined in a tag-based language.

17  
18 9. (Original) The method of claim 1, wherein the one or more  
19 manifests are defined in extensible markup language (XML).

20  
21 10. (Original) The method of claim 1, wherein the network comprises  
22 the Internet.

1 11. (Original) One or more computer-readable media comprising  
2 computer-readable instructions thereon which, when executed by a computer,  
3 cause the computer to implement the method of claim 1.

4  
5 12. (Original) A computer programmed with instructions which, when  
6 executed by the computer, implemented the method of claim 1.

7  
8 13. (Currently Amended) One or more computer-readable media  
9 comprising computer-readable instructions thereon which, when executed by a  
10 computer, cause the computer to:

11 describe one or more software extensions using extensible markup  
12 language (XML), the extensions being configured for incorporation on a client,  
13 said describing defining a manifest containing at least one list of files comprising  
14 an extension, the manifest being configured to assist in one or more of the  
15 following: organizing delivery of individual files listed in the manifest, validating  
16 individual files listed in the manifest, and updating individual files listed in the  
17 manifest; and

18 deliver the manifest to the client via the network, at least some of the  
19 extensions being downloadable by streaming extension files to the client in a  
20 manner that enables a user to begin to interact with the extension sooner than if the  
21 user had to wait for the entire extension to load.

22  
23 14. (Currently Amended) A method for receiving software via a network  
24 comprising:  
25

1 receiving a manifest that contains at least one list of files comprising a  
2 software extension that is to be downloaded via a network and incorporated on a  
3 client, the manifest being defined in extensible markup language (XML), the  
4 manifest being configured to assist in:

5 organizing delivery of the files,

6 validating individual files listed in the manifest, and

7 updating individual files listed in the manifest; and

8 downloading files from the list of files contained in the manifest;

9 wherein the extension is downloadable by streaming extension files to the  
10 client in a manner that enables a user to begin to interact with the extension sooner  
11 than if the user had to wait for the entire extension to load.

12  
13 15. (Original) The method of claim 14, wherein the software extension  
14 is to be incorporated into a software platform executing on the client.

15  
16 16. (Original) The method of claim 14, wherein the downloading of the  
17 files takes place by downloading the files in an order that is described in the  
18 manifest.

19  
20 17. (Original) The method of claim 14 further comprising validating one  
21 or more downloaded files using the manifest.

22  
23 18. (Original) The method of claim 14 further comprising updating one  
24 or more files using the manifest.

1 19. (Original) One or more computer-readable media comprising  
2 computer-readable instructions thereon which, when executed by a computer,  
3 cause the computer to implement the method of claim 14.

4  
5 20. (Original) A computer programmed with instructions which, when  
6 executed by the computer, implement the method of claim 14.

7  
8 21. (Currently Amended) A data structure embodied on a computer-  
9 readable medium ~~and configured to assist in delivering software extensions via the~~  
10 ~~Internet~~, the data structure comprising:

11 a list of one or more files that are utilized in a software extension that is  
12 configured to extend a software application executing on a client;

13 one or more hashes each of which being associated with a particular listed  
14 file; and

15 one or more file groups, individual files being associated with individual  
16 file groups, the file groups determining when particular files of the extension get  
17 downloaded to the client;

18 the data structure being configured to assist in delivering software  
19 extensions via the Internet.

20  
21 22. (Original) The data structure of claim 21, wherein the file groups  
22 determine where files are stored on the client.

23  
24 23. (Original) The data structure of claim 21, wherein the file groups  
25 determine how files are packaged for delivery.

1  
2 24. (Original) The data structure of claim 21, wherein the file groups  
3 determine where files are stored on the client and how files are packaged for  
4 delivery.

5  
6 25. (Original) The data structure of claim 21, wherein the file groups  
7 identify files that are to be downloaded before any other files.

8  
9 26. (Original) The data structure of claim 21, wherein the file groups  
10 identify files that are to be downloaded for offline use.

11  
12 27. (Original) The data structure of claim 21, wherein the file groups  
13 identify files that are only downloaded when they are required for the first time by  
14 a user.

15  
16 28. (Original) The data structure of claim 21, wherein the file groups  
17 identify files that are downloaded on demand and provide content that is available  
18 only when a user is online.

19  
20 29. (Original) The data structure of claim 21, wherein the file groups  
21 indicate file download priority.

22  
23 30. (Original) The data structure of claim 21, wherein the one or more  
24 hashes are used for security.  
25

1           31.    (Original) The data structure of claim 21, wherein the one or more  
2 hashes are used for versioning.

3  
4           32.    (Original) The data structure of claim 21, further comprising a  
5 storage size that is associated with an amount of storage required by the extension.

6  
7           33.    (Original) The data structure of claim 21, further comprising one or  
8 more class identifiers for individual dynamic link libraries (DLLs).

9  
10          34.    (Original) The data structure of claim 21, further comprising one or  
11 more dynamic link library (DLL) load dependencies.

12  
13          35.    (Original) The data structure of claim 21, further comprising:  
14 a storage size that is associated with an amount of storage required by the  
15 extension;

16          one or more class identifiers for individual dynamic link libraries (DLLs);  
17 and

18          one or more dynamic link library (DLL) load dependencies.

19  
20          36.    (Original) The data structure of claim 21 embodied as an extensible  
21 markup language (XML) file.

22  
23          37.    (Currently Amended) A method of providing software via a network  
24 comprising:  
25

1 describing one or more software extensions using one or more extensible  
2 markup language (XML) files, the extensions being configured for incorporation  
3 in a software program executing on a client, individual XML files providing  
4 individual manifests that contain a list of files that comprise an extension; and

5 storing the XML files in a Web-accessible location;

6 wherein at least some of the extensions are downloadable by streaming  
7 extension files to the client in a manner that enables a user to begin to interact with  
8 the extension sooner than if the user had to wait for the entire extension to load.

9  
10 38. (Original) The method of claim 37 further comprising storing  
11 extension files associated with the XML files in a Web-accessible location.

12  
13 39. (Original) The method of claim 37 further comprising:  
14 storing extension files associated with the XML files in a Web-accessible  
15 location; and

16 providing one or more XML files and one or more associated extension  
17 files to a client via the network.

18  
19 40. (Original) The method of claim 37, wherein individual manifests can  
20 contain one or more file hashes that can be used for file security.

21  
22 41. (Original) The method of claim 37, wherein individual manifests can  
23 contain one or more file hashes that can be used for versioning.



1 42. (Original) The method of claim 37, wherein individual manifests  
2 comprise one or more file groups that determine when particular files are  
3 downloaded to the client.

4  
5 43. (Original) The method of claim 42, wherein the file groups  
6 determine where files are stored on the client.

7  
8 44. (Original) The method of claim 42, wherein the file groups  
9 determine how files are packaged.

10  
11 45. (Original) The method of claim 42, wherein the file groups  
12 determine where files are stored on the client and how files are packaged.

13  
14 46. (Original) One or more computer-readable media having computer-  
15 readable instructions thereon which, when executed by a computer, cause the  
16 computer to implement the method of claim 37.

17  
18 47. (Original) A security method for downloading software extensions  
19 via the Internet comprising:

20 receiving, via the Internet, a package manifest containing a list of multiple  
21 files that comprise a software extension that is to be incorporated into an  
22 application program executing on a client, the list containing a hash for one or  
23 more of the files comprising the software extension;

24 receiving, via the Internet, the multiple files that are described in the  
25 package manifest;

1 creating a hash for one or more of the multiple received files; and  
2 comparing the created hash of the one or more files with corresponding file  
3 hashes contained in the package manifest to ascertain whether one or more of the  
4 received file is secure.

5  
6 48. (Original) The security method of claim 47, wherein the package  
7 manifest comprises an XML file.

8  
9 49. (Original) One or more computer-readable media comprising  
10 computer-readable instructions thereon which, when executed by a computer,  
11 cause the computer to implement the method of claim 47.

12  
13 50. (Original) One or more client computers programmed with  
14 instructions which, when executed by the one or more computers, cause the one or  
15 more computers to implement the method of claim 47.

16  
17 51. (Original) An updating method for updating software extensions via  
18 the Internet comprising:

19 receiving, via the Internet, a package manifest containing a list of multiple  
20 files that comprise a newer version of a software extension that is to be  
21 incorporated into an application program executing on a client that contains an  
22 older software extension version, the list containing a hash for one or more of the  
23 files comprising the newer version of the software extension;

24 comparing one or more hashes that are received with one or more hashes of  
25 files from the older version of the software extension;

1 for any hashes of corresponding files from the different versions that are  
2 different, downloading a new file from a web server; and

3 for any hashes of corresponding files from the different versions that are the  
4 same, copying a file from an old local directory on the client to a new local  
5 directory on the client associated with the newer version of the extension.

6  
7 52. (Original) The updating method of claim 51, wherein the package  
8 manifest comprises an XML file.

9  
10 53. (Original) One or more computer-readable media comprising  
11 computer-readable instructions thereon which, when executed by a computer,  
12 cause the computer to implement the method of claim 51.

13  
14 54. (Original) One or more client computers programmed with  
15 instructions which, when executed by the one or more computers, cause the one or  
16 more computers to implement the method of claim 51.

17  
18 55. (Original) A data structure embodied on a computer-readable  
19 medium comprising:

20 one or more first tags indicative of associated file groups associated with an  
21 Internet-downloadable software extension that can extend an application program  
22 executing on a client; and

23 one or more second tags indicative of specific files that comprise the  
24 software extension.  
25

1           56.    (Original) The data structure of claim 55, wherein the one or more  
2 second tags can contain hashes for individual files.

3  
4           57.    (Original) The data structure of claim 55 further comprising one or  
5 more third tags indicative of file load dependencies.

6  
7           58.    (Original) The data structure of claim 55 further comprising one or  
8 more third tags indicative of COMClass IDs.

9  
10          59.    (Original) The data structure of claim 55 further comprising:  
11 one or more third tags indicative of file load dependencies; and  
12 one or more fourth tags indicative of COMClass IDs.

13  
14          60.    (Original) The data structure of claim 55, wherein the ordering of the  
15 file groups implicitly defines a download order of the files.

16  
17          61.    (Original) The data structure of claim 55, wherein the tags comprise  
18 XML tags.

19  
20          62.    (Cancelled).

21  
22          63.    (Cancelled).

23  
24          64.    (Original) A queue management method comprising:  
25

1 defining a download queue that controls when files are to be downloaded to  
2 a client, the files pertaining to a software extension that is to be incorporated into  
3 an application program executing on the client;

4 ascertaining whether a user action at the client requires one or more files  
5 that are not currently being downloaded; and

6 manipulating the download queue responsive to a user action that requires  
7 one or more files that are not currently being downloaded so that the one or more  
8 required files are downloaded sooner than they would otherwise be.

9  
10 65. (Original) The queue management method of claim 64, wherein the  
11 download queue contains multiple package objects each of which contains a list of  
12 one or more files that correspond to a software extension, and said manipulating  
13 comprises moving a package object to the head of the download queue.

14  
15 66. (Original) The queue management method of claim 65, wherein said  
16 ascertaining comprises determining whether a required file is associated with a  
17 package object whose files are currently being downloaded.

18  
19 67. (Original) The queue management method of claim 65, wherein said  
20 manipulating comprises:

21 ascertaining an identifier associated with a requested file; and

22 locating a package object with a corresponding identifier.  
23  
24  
25

1           68. (Original) One or more computer-readable media comprising  
2 computer-readable instructions thereon which, when executed by a computer,  
3 cause the computer to implement the method of claim 64.

4  
5           69. (Currently Amended) A method of creating software packages for  
6 delivery via the Internet comprising:

7           identifying end user features;

8           identifying shared dependencies between the end user features;

9           creating individual software packages for the end user features;

10          creating individual software packages for the shared dependencies; and

11          hosting the software packages on a web server;

12          wherein both of said acts of identifying and both of said acts of creating  
13 provide software extensions that are created in a uniform manner independent of  
14 end user input.

15  
16          70. (Original) The method of claim 69, wherein the packages comprise  
17 one or more files that are associated with a single application program that  
18 provides multiple different functionalities, and the packages extend, in some way,  
19 the functionalities that are provided by the single application program.

20  
21          71. (Currently Amended) An automated software tool comprising a  
22 package manifest creation tool configured to:

23          receive one or more input parameters pertaining to a package manifest that  
24 is to describe a software extension that is configured to extend a software  
25 application executing on a client; and

1 generate a package manifest that describes the extension, the package  
2 manifest being generated using a hierarchical language;

3 wherein the extension is downloadable by streaming extension files to the  
4 client in a manner that enables a user to begin to interact with the extension sooner  
5 than if the user had to wait for the entire extension to load.

6  
7 72. (Original) The automated software tool of claim 71, wherein the  
8 hierarchical language comprises a tag-based language.

9  
10 73. (Original) The automated software tool of claim 71, wherein the  
11 hierarchical language comprises extensible markup language (XML).

12  
13 74. (Original) The automated software tool of claim 71, wherein one  
14 input parameter specifies a directory containing files that are to be described in the  
15 package manifest.

16  
17 75. (Original) The automated software tool of claim 71, wherein one  
18 input parameter comprises file group information and load dependencies.

19  
20 76. (Original) The automated software tool of claim 71, wherein one  
21 input parameter comprises file usage statistics.

22  
23 77. (Original) The automated software tool of claim 76, wherein the file  
24 usage statistics are ascertained from scenario runs.

1 78. (Original) The automated software tool of claim 77, wherein  
2 individual scenarios have individual priorities.

3  
4 79. (Original) The automated software tool of claim 76, wherein the file  
5 usage statistics are ascertained from scenario runs that are collected by running  
6 logs on various scenarios.

7  
8 80. (Original) The automated software tool of claim 76, wherein the file  
9 usage statistics are ascertained from scenario runs that are collected by running  
10 logs on various scenarios, the scenarios having individual checkpoints that  
11 separate one scenario from another.

12  
13 81. (Currently Amended) The automated software tool of claim 76,  
14 wherein the file usage statistics are ~~ascertaining~~ ascertained dynamically by  
15 building a knowledge base that describes tasks that users typically accomplish.

16  
17 82. (Original) A method for creating a package manifest comprising:  
18 receiving information pertaining to one or more extension directories that  
19 contain files that comprise a software extension that is to be incorporated into a  
20 software application program to extend the application program;  
21 receiving, if any, information pertaining to file groups or load  
22 dependencies;  
23 receiving, if any, information pertaining to file usage statistics; and  
24 generating a package manifest based on the received information.  
25



1           83.    (Original) The method of claim 82, wherein the package manifest is  
2 generated in XML.

3  
4           84.    (Original) The method of claim 82, wherein the file usage statistics  
5 can be provided based upon scenario runs.

6  
7           85.    (Original) One or more computer-readable media having computer-  
8 readable instructions thereon which, when executed by a computer, cause the  
9 computer to implement the method of claim 82.

10  
11           86.    (Original) A method of providing software extensions via the  
12 Internet comprising:

13                assigning one or more files to one or more scenarios to provide multiple  
14 different scenarios that describe ways that a user interacts with a software  
15 application program;

16                assigning a priority to each of the scenarios;

17                sorting multiple files in accordance with their scenario priority or priorities;

18 and

19                downloading sorted files in an order defined by said sorting.

20  
21           87.    (Original) The method of claim 86 further comprising sorting the  
22 multiple files in accordance with one or more file groups.

23  
24           88.    (Original) The method of claim 86 further comprising sorting the  
25 multiple files in accordance with a file usage order within one or more scenarios.

1  
2 89. (Original) The method of claim 86 further comprising:  
3 sorting the multiple files in accordance with one or more file groups; and  
4 sorting the multiple files in accordance with a file usage order within one or  
5 more scenarios.

6  
7 90. (Original) One or more computer-readable media having computer-  
8 readable instructions thereon which, when executed by a computer, cause the  
9 computer to implement the method of claim 86.

10  
11 91. (Original) A method of ordering files for download to a client  
12 comprising:

13 sorting multiple files by one or more file groups;  
14 sorting the multiple files based on scenario priority of one or more  
15 scenarios into which each file can be placed;  
16 sorting the multiple files by file usage order within one or more scenarios.

17  
18 92. (Original) The method of claim 91 further comprising determining a  
19 file group from a manifest defined in XML.

20  
21 93. (Original) The method of claim 91 further comprising if no file  
22 group information is provided, assuming that all files of a particular type are of a  
23 higher priority than other files of different types.

1           94.    (Original) The method of claim 91, wherein said sorting by file  
2 usage order comprises sorting the files according to the average order in which the  
3 files were downloaded within their particular priority or priorities.

4  
5           95.    (Original) One or more computer-readable media having computer-  
6 readable instructions thereon which, when executed by a computer, cause the  
7 computer to implement the method of claim 91.

8  
9           96.    (Original) One or more server computers programmed with  
10 instructions which, when executed by the one or more server computers, cause the  
11 one or more server computers to implement the method of claim 91.

---